

文章编号: 1001-0920(2005)06-0702-04

## 基于连接识别的协同进化种群分割算法研究

孙晓燕, 巩敦卫, 杜学艳

(中国矿业大学 信息与电气工程学院, 江苏 徐州 221008)

**摘要:** 针对已有协同进化算法种群分割方法的不足, 借鉴连接识别思想, 提出用分段间隔相关迭代连接识别进行种群分割的算法, 并给出算法的思想、步骤及具体实施方案, 将其应用于典型的复杂函数优化问题, 简要分析算法在搜索效率、计算复杂性等方面的性能以及相关参数对算法性能的影响, 说明算法的可行性

**关键词:** 协同进化遗传算法; 连接识别; 种群分割

**中图分类号:** TP18 **文献标识码:** A

## On linkage identification based coevolutionary population decomposition algorithm

SUN Xiao-yan, GONG Dun-wei, DU Xue-yan

(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008, China Correspondent: SUN Xiao-yan, E-mail: xysun78@126.com)

**Abstract:** Based on the idea of linkage identification, piecewise interval correlation by iteration (PIC) linkage identification based population decomposition algorithm is presented in allusion to the deficiency of existing population decomposition methods in coevolutionary algorithm. The methodology, steps and strategies of population decomposition are given in detail. The performance in search efficiency, computation complexity of the algorithm is analyzed by optimizing benchmark complicated functions and the effect of pertinent parameters on its performance is also analyzed.

**Key words:** coevolutionary genetic algorithm; linkage identification; population decomposition

### 1 引言

合作型协同进化算法主要是针对复杂协同适应性问题提出的一类进化结构<sup>[1]</sup>, 已广泛应用于多变量复杂函数的优化<sup>[2]</sup>、管理系统货存量的优化、神经网络拓补结构和权值的优化<sup>[3]</sup>、模糊控制系统中模糊规则及隶属度函数的确定<sup>[4]</sup>等方面

合作型协同进化算法的思想是对待优化的变量进行分组, 将一个复杂多变量系统转化为多个简单的少变量系统, 对分组后的多个少变量系统分别编码, 形成多个独立的子种群, 各种群独立进化, 这就是种群分割。这样子种群的规模远远小于原种群规

模, 且大大缩短了编码长度, 加快收敛速度。由此可见, 合作型协同进化算法的一个关键技术就是种群分割。但是对于种群分割这一问题, 其理论研究并不充分。

针对已有协同进化算法种群分割方法的不足, 借鉴连接识别思想, 本文提出一种用分段间隔相关迭代连接识别进行种群分割的算法, 给出算法的思想、步骤及具体实施方案, 将其应用于典型复杂函数的优化, 简要分析了算法在搜索效率、计算复杂性等方面的性能以及相关参数对算法性能的影响, 说明该算法是可行的。

收稿日期: 2004-08-05; 修回日期: 2004-10-08

基金项目: 国家自然科学基金项目(60304016)。

作者简介: 孙晓燕(1978—), 女, 江苏丰县人, 硕士生, 从事进化计算等研究; 巩敦卫(1970—), 男, 江苏铜山人, 教授, 博士生导师, 从事智能控制、进化计算等研究

## 2 基于概率模型的连接识别算法

概率模型连接识别算法通过种群个体分布情况获得变量之间的连接信息<sup>[5]</sup>。本文考虑实数编码方式, 设变量  $x$  的第  $i$  个和第  $j$  个分量  $x_i, x_j$  在种群中各个体基因座上的值分别为  $\tilde{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,N}\}$  和  $\tilde{x}_j = \{x_{j,1}, x_{j,2}, \dots, x_{j,N}\}$ , 则分量  $x_i, x_j$  之间的相关系数为<sup>[6]</sup>

$$\rho_{ij} = \frac{\frac{1}{N} \sum_{p=1}^N (x_{i,p} - \bar{x}_i)(x_{j,p} - \bar{x}_j)}{\sqrt{V(\tilde{x}_i)V(\tilde{x}_j)}} \quad (1)$$

其中:  $\bar{x}_i, \bar{x}_j$  分别为  $\tilde{x}_i, \tilde{x}_j$  的平均值;  $V(\tilde{x}_i), V(\tilde{x}_j)$  分别为  $\tilde{x}_i, \tilde{x}_j$  的方差  $\rho_{ij}$  大小反映了分量  $x_i, x_j$  之间的相关性, 若  $\rho_{ij}$  较大, 则认为两个分量之间存在较强的连接关系, 故这里定义  $\rho_{ij}$  值为分量  $x_i, x_j$  之间的连接强度。当分量个数较多且种群规模较大时, 采用式(1) 其计算量非常庞大, 因此为有效计算  $\rho_{ij}$ , 本文采用分段间隔相关迭代(PICD) 算法<sup>[6]</sup>。下面给出算法的具体实施步骤:

Step 1: 子空间划分

Step 1. 1: 将  $\tilde{x}_i$  和  $\tilde{x}_j$  构成的空间划分为 4 个相等子空间  $N_1, N_2, N_3$  和  $N_4$

Step 1. 2: 检测各子空间个体数, 若某子空间个体数少于一设定阈值  $N_{\min}$ , 则停止对该子空间的划分; 否则转 Step 1. 1, 在各子空间中继续 4 等分。

Step 2: 变量连接强度  $\rho_{ij}$  计算。从最小子空间开始, 倒退式往上一层空间迭代计算, 记各层每个 1/4 空间得到的连接强度值为  $\rho_s, s = 1, 2, 3, 4$

Step 2. 1: 若某子空间只有一个个体, 则  $\rho_s = 0$ ; 否则用式(1) 计算  $\rho_s$ 。

Step 2. 2: 计算上一层空间, 若至少存在一个子空间, 其中的个体数大于等于  $N_{\min}$ , 则对其每一个下层子空间的  $\rho_s$  取加权平均作为该层子空间连接强度值, 即

$$\rho_s = \frac{\sum_{s=1}^4 N_s * \rho_s}{\sum_{s=1}^4 N_s} \quad (2)$$

否则, 用式(1) 继续进行计算, 其中的  $N$  是所考虑子空间内的个体数

## 3 基于连接识别的种群分割算法

借鉴上述变量间连接识别的思想及确定变量间连接强度的方法, 根据变量间连接强度的大小对合作型协同进化优化变量进行分组, 实现有效的种群分割。设定连接强度阈值为  $K_p$ , 若  $\rho_{ij} > K_p$ , 则认为变量  $x_i$  和  $x_j$  存在紧连接关系, 分别将它们放入紧连接组中; 其余变量放入一组。针对变量分割与种群进化之间的相互关联性, 本文给出下述两种分割算法, 分别称为一次性连接识别分割算法和进化连接

识别分割算法

### 3.1 一次性连接识别分割算法(OLDA)

#### 3.1.1 分割步骤

Step 1: 初始化一种群, 进行  $T$  代标准遗传算法操作;

Step 2: 进化  $T$  代后, 运用 PICD 算法计算出所有变量间的相关系数, 得到相关系数矩阵  $C$ ;

Step 3: 根据变量间连接强度的大小进行种群分割, 分割依据下式进行:

$$\begin{aligned} & x_i \quad X_{\text{tight } s} \mid S \quad \{1, \dots, S\}, \\ & \text{if } \{\rho_{ij} > K_p \mid \forall x_j \in X_s\} \\ & \quad \{\rho_{ij} < K_p \mid \forall x_j \notin X_s\}, \end{aligned} \quad (3)$$

其中  $S$  为紧连接组的个数

#### 3.1.2 $T$ 和 $K_p$ 的取值

采用标准遗传算法进化  $T$  代, 目的是获得相对准确的个体分布信息, 实现变量间连接强度的计算。考虑计算复杂性和准确性,  $T$  取值要适中。若  $T$  取值过大或过小, 对于准确计算  $\rho_{ij}$  不利, 且会造成计算浪费, 因此  $T$  的取值需在仿真中反复试探。影响准确分组的一个重要因素便是分组阈值  $K_p$  的取值。若  $K_p$  取值较大, 则分组过滤性要求较高, 结果可能认为所有变量都没有连接关系; 反之则分组太粗, 有可能导致认为所有变量之间都有连接关系。这里取  $K_p = C_{\max} - 0.05$ , 其中  $C_{\max} = \max(C)$ 。显然  $K_p$  取值随进化而动态地改变。

### 3.2 进化连接识别分割算法(ELDA)

#### 3.2.1 分割步骤

Step 1: 初始化一种群, 进行  $T_1$  代标准遗传算法操作;

Step 2: 进化  $T_1$  代后, 运用 PICD 算法计算出所有变量之间的相关系数, 得到相关系数矩阵  $C$ ;

Step 3: 根据变量间连接强度的大小进行种群分割, 依据如式(3) 所示;

Step 4: 对分割后得到的子种群进行  $T_2$  代协同进化;

Step 5: 对于其余变量构成的连接组, 转 Step 2; 如此反复迭代, 直至变量分割完毕。

#### 3.2.2 阈值 $K_p$ 的设计

在进化连接识别分割算法中, 参数  $K_p$  亦需考虑 3.1.2 所述, 这里采用下述  $K_p$  取值方法:

$$K_p = 2 \arctan(t) / \pi$$

即随着时间  $t$  的增加,  $K_p$  值增加, 使得分组越来越精确。

## 4 实例验证

为验证本文算法的可行性, 将其应用于两类在合作协同进化领域常用的典型函数优化问题:

$$\min f_1(x) = 418\ 982\ 9n + \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}),$$

$$x_i \in [-500, 500], n = 10; \quad (4)$$

$$\min f_2(x) = \sum_{j=2,4,6} 100(x_j - x_i^2)^2 + (1 - x_i^2)^2, x_i, x_j \in [-10, 10] \quad (5)$$

其中:  $f_1$  代表一类可分函数, 即变量间没有明显的耦合关系;  $f_2$  代表一类不可分函数, 即函数变量之间存在明显的耦合关系。函数  $f_1$  在  $x = (-420\ 968\ 7, -420\ 968\ 7, \dots)$  处取得全局最小值 0, 函数  $f_2$  在  $x = (1, 1, \dots)$  处取得全局最小值 0。

图 1 为函数  $f_1$  在  $n = 2$  时的图形, 可知该函数存在大量的局部极小, 在  $x = (303, 303, \dots)$  处取得局部最优, 显然该函数局部最优离其全局最优较远, 采用一般优化算法极易陷入局部最优, 故采用这个函数可以衡量本文算法的搜索性能。另外为增加寻优难度, 变量个数  $n = 10$ , 以充分说明算法的有效性。

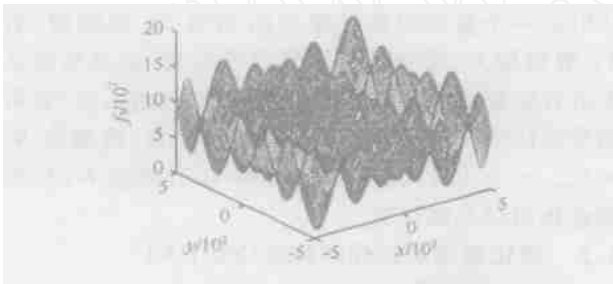


图 1  $n = 2$  时函数  $f_1$  的图形

图 2 为函数  $f_2$  在  $i = 1, j = 2$  时的图形。为增加寻优难度, 将变量个数扩展到  $n = 6$ , 以有力说明本文算法在变量分割上的有效性。

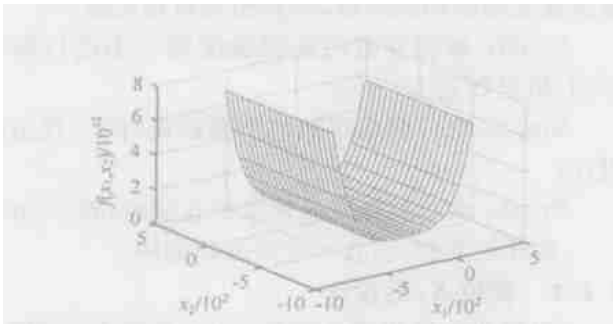


图 2  $i = 1, j = 2$  时函数  $f_2$  的图形

将本文提出的一次性连接识别分割算法 (OL DA) 和进化连接识别分割算法 (EL DA) 与合作型协同进化一般分割算法 (GDA) 在算法优化的质量、计算复杂性即函数评价次数两个方面进行比较。为消除一次运行结果中随机的影响, 本文对所有的算法均运行 20 次。

各算法所需相关参数取值见表 1 和表 2。其中本文提出的两种种群分割方法子种群个数指的是连接识别后得到的最终子种群个数, 各子种群优化的变量即为识别后的分组。为便于比较, 设定 GDA 子种群个数为 3, 其优化变量是等概率地从所有变量中随机选择。对于 OL DA, 连接识别前一般遗传算法进化代数  $T$  的取值见下述分析。

表 1 函数  $f_1$  仿真相关参数取值

算法	EL DA	OL DA	GDA
交叉概率	0.95	0.95	0.95
变异概率	0.09	0.09	0.09
子种群个数	3	3	3
种群规模	200	200	200
停滞阈值 $t$	350	350	350
停滞阈值 $\delta$	0.1	0.1	0.1

表 2 函数  $f_2$  仿真相关参数取值

算法	EL DA	OL DA	GDA
交叉概率	0.95	0.95	0.95
变异概率	0.09	0.09	0.09
子种群个数	3	3	3
种群规模	100	100	100
停滞阈值 $t$	100	100	100
停滞阈值 $\delta$	3	3	3

以相同协同进化代数  $T_c$  作为终止条件; 设定相同的协同进化代数  $T_c$  (见表 1, 表 2 第 5 行), 比较 3 种算法搜索到的最优解, 运行平均结果见表 3。

表 3 相同协同进化代数下各算法得到的最优解

算法	函数 $f_1$	函数 $f_2$
OL DA	0.055	0.18
EL DA	0.035	0.10
GDA	0.055	1.26

以  $|f_d - f| \leq \delta$  为终止条件; 其中:  $f$  为各算法实际搜索到的最优解,  $f_d$  为期望最优解,  $\delta$  为停滞阈值。因为本文优化的两类函数最优值均为 0, 所以实际终止条件为  $|f| \leq \delta$  ( $\delta$  取值见表 1, 表 2 最后一行)。

以搜索到相同满意解为终止条件, 比较 3 种算法的协同进化代数, 以比较其计算复杂性, 运行结果见表 4。

表 4 相同最优解下各算法的协同进化代数

算法	函数 $f_1$	函数 $f_2$
OL DA	150.5	30.2
EL DA	159.3	24.5
GDA	150	55.6

从表 3 和 4 可知, 对于可分函数, 上述 3 种算法

在优化解的质量和计算复杂性方面, 其性能都是相当的。而对于变量间含有耦合关系的函数, 在相同进化代数的前提下, 采用进化连接识别合作型协同进化算法搜索到最优解的质量最好, 一次性连接识别合作型协同进化算法次之, 一般合作型协同进化算法搜索到最优解的质量较差。在相同精度要求的前提下, 采用进化连接识别合作型协同进化算法搜索到最优解时协同进化的代数最少, 一次性连接识别合作型协同进化算法次之, 一般合作型协同进化算法的协同进化代数最多, 这表明了本文算法的有效性。

现在讨论  $T$  对算法性能的影响: 连接识别前进化代数  $T$  对 OL DA 性能影响结果见表 5 和表 6。表 6 中分组成功率指不可分函数  $f_2$  (其正确分组应该是  $x_1, x_2; x_3, x_4; x_5, x_6$  三组) 20 次实验可以成功分组的概率。在仿真过程中发现, 对于可分函数,  $T$  取值对算法性能的影响很微弱, 表 5 也证明了这一点。其原因在于对于可分函数, 变量之间没有明显连接关系, 因此无论怎样分组都不会影响其相对关系, 从而不会影响算法性能。对于不可分函数,  $T$  取值对算法性能影响较大, 从表 6 可以看出, 在  $T = 10$  时算法的性能是最好的。其原因正如 3.1.2 所述, 不正确的分组会破坏变量之间的相对关系, 使得各子种群之间存在强连接关系, 导致算法性能变差。关于 EL DA 中  $T_1$  和  $T_2$  取值对算法性能的影响与前类似, 不再赘述。

表 5 优化函数  $f_1$  时  $T$  对 OL DA 的影响

$T$	最优解	迭代次数
5	0.095	150.5
10	0.100	159.7
15	0.098	158.3

表 6 优化函数  $f_2$  时  $T$  对 OL DA 的影响

$T$	最优解	迭代次数	分组成功率/%
5	0.32	279.9	58
10	0.13	210.5	81
15	0.21	231.3	62

综上所述, 若事先不知道待优化对象变量之间是否存在耦合关系, 或存在什么样的耦合关系时, 采

用本文给出的算法对其变量进行分组是完全可行的。当然, 连接识别的过程需要一定计算量, 在目前的计算机条件下这是不成问题的, 故本文算法对于合作型协同进化算法种群分割方法具有较好的理论指导意义。

## 5 结 语

针对协同进化算法关键技术——种群分割方法存在的不足, 本文借鉴连接识别思想, 提出用分段间隔相关迭代连接识别进行种群分割的算法, 给出了算法的思想、步骤及具体实施方案。基于典型复杂函数的优化, 简要分析了算法在搜索效率、计算复杂性等方面的性能及相关参数对算法性能的影响, 说明算法的可行性, 为协同进化种群分割算法提供了理论指导。用分段间隔相关迭代连接识别设计合作型协同进化算法中代表个体选择是进一步要研究的内容。

## 参考文献 (References)

- [1] Potter M A. The design and analysis of a computational model of cooperative coevolution [D]. Fairfax: George Mason University, 1997.
- [2] Mitchell A Potter, De Jong K A. A cooperative coevolutionary approach to function optimization [A]. *Proc of the 3rd Conf of Parallel Problem Solving from Nature* [C]. Jerusalem: 1994: 249-257.
- [3] Garcia-Pedrajas N, Hervás-Martínez C, Muñoz-Pérez J. Multi-objective cooperative coevolution of artificial neural networks [J]. *Neural Networks*, 2002, 15 (10): 1259-1278.
- [4] Carlos Andres Pena-Reyes, Moshe Sipper. Fuzzy CoCo: A cooperative coevolutionary approach to fuzzy modeling [J]. *IEEE Transactions on Fuzzy Systems*, 2001, 9 (5): 727-737.
- [5] Shigeyoshi Tsutsui, David E Goldberg, Kumara Sastry. Progress toward linkage identification in real-coded gas with simplex crossover [R]. Illinois Genetic Algorithm Laboratory, 2000.
- [6] Shigeyoshi Tsutsui, David E Goldberg. Simplex crossover and linkage identification: Single-stage evolution vs multi-stage evolution [R]. Illinois Genetic Algorithm Laboratory, 2002.